

wireless telephone; (3) diverting to a temporary storage within the mobile device or wireless telephone, an audio content stream received at an audio input; (4) storing the audio content stream within the mobile device or wireless telephone in a format that is compatible with adding audio content to the electronic message; and (5) attaching the formatted audio content to the electronic message at the mobile device or wireless telephone. An alternate embodiment of the method included (1) initiating the creation of an electronic message at a mobile device, the electronic message to include audio content; (2) capturing audio content at the mobile device from an audio content stream being received at an audio input, wherein the audio content stream is generated by a user speaking into the audio input; and (3) adding the audio content to the electronic message at the mobile device in a format that is compatible with the electronic message.

2. The source code attached as Exhibit A shows software for a wireless telephone as evidence of our reduction to practice. This source code represents an initial working implementation of the invention, including the features identified above, for a wireless telephone. Specifically, the source code shows starting a recording, stopping a recording, and inserting the recording as an attachment.

3. The source code differences attached as Exhibit B show source code changes as evidence of our reduction to practice. The source code differences were produced when the initial working implementation of the invention (the source code attached as Exhibit A), including the features identified above, was checked-in to the source code tree for the wireless telephone project.

4. The difference log attached as Exhibit C shows a development history of the source code for the wireless telephone as evidence of our reduction to practice. The difference log shows the initial working implementation of the invention, including the features identified above, as well as further refinements, including "Detect record failure," "Delete existing wav file if necessary," and "cancel the recording." Relevant entries have been identified with an arrow in the left margin.


5. The transcript attached as Exhibit D shows a demonstration, performed by Mr. Ferrell for Bill Gates during a keynote address given by Bill Gates at a telecommunication conference, of a wireless telephone prototype as evidence of our reduction to practice. As described in the transcript, Mr. Ferrell demonstrated using the wireless telephone in responding to an email. Mr. Ferrell selected a "Reply by voice" option, recorded a voice reply, and sent the voice reply as an attachment to a reply email. In composing the electronic message, the prototype wireless telephone operated in accordance with the foregoing methods, and therefore represents a working prototype of our reduction to practice, including each of the features identified above.

6. Although dates and certain other information have been redacted from the evidence attached as Exhibits A, B, C & D, our reduction to practice occurred at least as early as January 9, 2000.

7. Our reduction to practice occurred within the United States of America, as evidenced by checking-in the source code to a source code tree, and the manufacture of the wireless telephone prototype that was used in the demonstration performed by Mr. Ferrell, within the United States of America. The telecommunication conference keynote address (and corresponding demonstration) occurred in Geneva Switzerland (a WTO Member Country since 1 Jan 1995).

I hereby declare under penalty of perjury under the laws of the United States of America that all statements made herein are of my own knowledge and are true; and further, that these statements are made with the knowledge that willful false statements, and the like so made, are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated this 19 day of December, 2004



Neil R. N. Enns

Dated this _____ day of December, 2004

John Ian Ferrell

wireless telephone; (3) diverting to a temporary storage within the mobile device or wireless telephone, an audio content stream received at an audio input; (4) storing the audio content stream within the mobile device or wireless telephone in a format that is compatible with adding audio content to the electronic message; and (5) attaching the formatted audio content to the electronic message at the mobile device or wireless telephone. An alternate embodiment of the method included (1) initiating the creation of an electronic message at a mobile device, the electronic message to include audio content; (2) capturing audio content at the mobile device from an audio content stream being received at an audio input, wherein the audio content stream is generated by a user speaking into the audio input; and (3) adding the audio content to the electronic message at the mobile device in a format that is compatible with the electronic message.

2. The source code attached as Exhibit A shows software for a wireless telephone as evidence of our reduction to practice. This source code represents an initial working implementation of the invention, including the features identified above, for a wireless telephone. Specifically, the source code shows starting a recording, stopping a recording, and inserting the recording as an attachment.

3. The source code differences attached as Exhibit B show source code changes as evidence of our reduction to practice. The source code differences were produced when the initial working implementation of the invention (the source code attached as Exhibit A), including the features identified above, was checked-in to the source code tree for the wireless telephone project.

4. The difference log attached as Exhibit C shows a development history of the source code for the wireless telephone as evidence of our reduction to practice. The difference log shows the initial working implementation of the invention, including the features identified above, as well as further refinements, including "Detect record failure," "Delete existing wav file if necessary," and "cancel the recording." Relevant entries have been identified with an arrow in the left margin.

5. The transcript attached as Exhibit D shows a demonstration, performed by Mr. Ferrell for Bill Gates during a keynote address given by Bill Gates at a telecommunication conference, of a wireless telephone prototype as evidence of our reduction to practice. As described in the transcript, Mr. Ferrell demonstrated using the wireless telephone in responding to an email. Mr. Ferrell selected a "Reply by voice" option, recorded a voice reply, and sent the voice reply as an attachment to a reply email. In composing the electronic message, the prototype wireless telephone operated in accordance with the foregoing methods, and therefore represents a working prototype of our reduction to practice, including each of the features identified above.

6. Although dates and certain other information have been redacted from the evidence attached as Exhibits A, B, C & D, our reduction to practice occurred at least as early as January 9, 2000.

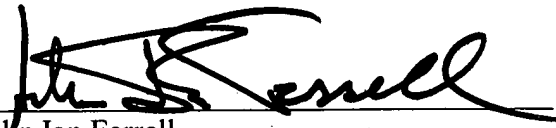
7. Our reduction to practice occurred within the United States of America, as evidenced by checking-in the source code to a source code tree, and the manufacture of the wireless telephone prototype that was used in the demonstration performed by Mr. Ferrell, within the United States of America. The telecommunication conference keynote address (and corresponding demonstration) occurred in Geneva Switzerland (a WTO Member Country since 1 Jan 1995).

I hereby declare under penalty of perjury under the laws of the United States of America that all statements made herein are of my own knowledge and are true; and further, that these statements are made with the knowledge that willful false statements, and the like so made, are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated this _____ day of December, 2004

Dated this 27 day of December, 2004

Neil R. N. Enns



John Ian Ferrell

EMK0000000202V001.DOC

```
HRESULT CMainWnd::StartRecording(CMainWnd* pMainWnd, HWND hDlg)
{
    RECT rcDlg;

    CM_VOICE_RECORDER vrec = {0};

    HRESULT hr = E_FAIL;

    lstrcpy(pMainWnd->m_szAttachment, TEXT("\\Windows\\Message.wav"));

    // If the file already exists, delete it.
    if(GetFileAttributes(pMainWnd->m_szAttachment) != -1)
    {
        RETAILMSG(1, (TEXT("CMainWnd StartRecording: Delete existing wave file\n")));

        DeleteFile(pMainWnd->m_szAttachment);
    }

    vrec.cb          = sizeof(vrec);
    vrec.dwStyle     = 0; // BUGBUG - Depending on the button msgs we get, perhaps modal
    is better?
    vrec.hwndParent  = hDlg;
    vrec.id          = IDC_MSG_VOICE;

    vrec.xPos = vrec.yPos = -1; // Center

    vrec.lpszRecordFileName = pMainWnd->m_szAttachment;

    if((pMainWnd->m_hWndVoice = VoiceRecorder_Create(&vrec)) == NULL)
    {
        RETAILMSG(1, (TEXT("CMainWnd StartRecording: VoiceRecorder_Create failed\n")));

        goto Error;
    }

    // Hide the window, so we can display our own progress bar.
    ShowWindow(pMainWnd->m_hWndVoice, SW_HIDE);

    // Put the progress bar at the bottom of the dlg. Move the body window
    // up a bit so the two windows don't overlap. We'll change the body window
    // to be back as it was once the recording is complete.

    GetClientRect(hDlg, &rcDlg);

    pMainWnd->m_hWndProgress = CreateWindow(PROGRESS_CLASS, NULL, WS_CHILD,
                                           0, rcDlg.bottom - 20, rcDlg.right, 20,
                                           hDlg, (HMENU)IDC_PROG_BAR, ghInst, NULL);

    ResizeBodyWindow(pMainWnd, hDlg, 25);

    ShowWindow(pMainWnd->m_hWndProgress, SW_SHOW);

    // Start the timer now for updating the progress bar.
    SetTimer(hDlg, RECORD_TIMER_ID, 1000, NULL);

    // Kick off the recording!
    ::SendMessage(pMainWnd->m_hWndVoice, VRM_RECORD, 0, 0);

    // Add a "Stop" btn to the soft keys.
    LoadSoftKeys(hDlg, IDI_STOP, TEXT("Stop"), 0, NULL, FALSE);

    // Everything worked ok.
    hr = S_OK;
}
```

Error:

```
    return hr;
}

HRESULT CMainWnd::StopRecording(CMainWnd* pMainWnd, HWND hDlg)
{
    BOOL fFailed;

    HRESULT hr = E_FAIL;

    ASSERT(pMainWnd->m_hWndVoice != NULL);

    // Kill any timer for updating a progress bar.
    KillTimer(hDlg, RECORD_TIMER_ID);

    // Now close our progress window.
    ASSERT(pMainWnd->m_hWndProgress != NULL);

    DestroyWindow(pMainWnd->m_hWndProgress);
    pMainWnd->m_hWndProgress = NULL;

    // Now close the voice control itself.
    ASSERT(pMainWnd->m_hWndVoice != NULL);

    // Make sure the msg body window is at a size that doesn't have to dodge the progress
    // bar.
    ResizeBodyWindow(pMainWnd, hDlg, 0);

    // We don't have a good attachment yet.
    pMainWnd->m_fGotAttachment = FALSE;

    // From the spec, "Stop Record/Playback save recording data if any and close Voice
    // Recorder"

    // Stop the recording!

    ::SendMessage(pMainWnd->m_hWndVoice, VRM_OK, 0, 0);

    // One would imagine that the voice control has destroyed the window.
    pMainWnd->m_hWndVoice = NULL;

    // Check the expected file exists.
    if((fFailed = (GetFileAttributes(pMainWnd->m_szAttachment) == -1)))
    {
        RETAILMSG(1, (TEXT("CMainWnd StopRecording: File %s does NOT exist\n"), pMainWnd->
        >m_szAttachment));

        // BUGBUG - For now, we don't care whether the final file got created or not.
        // For the real app, no file here means an error.
    }
    else
    {
        HANDLE hFile;

        // How big is the file? // BUGBUG - Make this debug only in the future.
        if((hFile = CreateFile(pMainWnd->m_szAttachment, GENERIC_READ, FILE_SHARE_READ,
        NULL,
        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL)) != INVALID_HANDLE_VALUE)
        {
            RETAILMSG(1, (TEXT("TMail: New file size is %d bytes\n"), GetFileSize(hFile,
            NULL)));

            CloseHandle(hFile);
        }
    }

    // Now add the attachment!
```



```

if(FAILED(hr = InsertAttachment(pMainWnd)))
{
    RETAILMSG(1, (TEXT("CMainWnd StopRecording: InsertAttachment failed %x\n"), hr));

    goto Error;
}

// At last, we have the attachment!
pMainWnd->m_fGotAttachment = TRUE;

// Everything worked ok.
hr = S_OK;

```

Error:

```

// Set the soft keys back to the default for a message being composed.
LoadSoftKeys(hDlg, IDI_SEND, TEXT("Send"), IDI_ARROWUP, TEXT("Options"), TRUE);

return hr;

```

```
#F main.cpp v31
#K text
#O in
#P 2.03.9006
#T 
#A GUYBARK4
#C First pass at Voice record
#I 32
#D 5487
26a27
> #include "voicectl.h"
155a157,159
>
>     BOOL      m_fGotAttachment; // TRUE if the msg has an attachment.
>     HWND      m_hWndVoice;      // Voice record window.
171c175
<     static HRESULT HandleAttach(CMainWnd* pMainWnd, HWND hDlg);
---
>     static HRESULT HandleVoice(CMainWnd* pMainWnd, HWND hDlg);
174c178,179
<     static HRESULT GetAttachment(CMainWnd* pMainWnd);
---
>     static HRESULT StartRecording(CMainWnd* pMainWnd);
>     static HRESULT StopRecording(CMainWnd* pMainWnd);
272a278,280
>
>     m_fGotAttachment = FALSE;
>     m_hWndVoice      = NULL;
1844c1852
<         pMainWnd->m_szAttachment[0] = '\0';
---
>         pMainWnd->m_fGotAttachment = FALSE;
1872c1880
<             if(FAILED(hr = HandleAttach(pMainWnd, hDlg)))
---
>             if(FAILED(hr = HandleVoice(pMainWnd, hDlg)))
1874c1882
<                 RETAILMSG(1, (TEXT("CMainWnd _NewMessageDlgProc: HandleAttach
failed %x\n"), hr));
---
>                 RETAILMSG(1, (TEXT("CMainWnd _NewMessageDlgProc: HandleVoice
failed %x\n"), hr));
1921a1930,1936
>         if(pMainWnd->m_hWndVoice != NULL)
>         {
>             ::SendMessage(pMainWnd->m_hWndVoice, VRM_CANCEL, 0, 0);
>
>             pMainWnd->m_hWndVoice = NULL;
>         }
2197c2212
<     if(pMainWnd->m_szAttachment[0] != '\0')
---
>     if(pMainWnd->m_fGotAttachment)
2845c2860
< HRESULT CMainWnd::HandleAttach(CMainWnd* pMainWnd, HWND hDlg)
---
> HRESULT CMainWnd::HandleVoice(CMainWnd* pMainWnd, HWND hDlg)
2853c2868
<     if(pMainWnd->m_szAttachment[0] != '\0')
---
>     if(pMainWnd->m_fGotAttachment)
2860c2875
<         RETAILMSG(1, (TEXT("CMainWnd HandleAttach: Out of memory\n")));
---
>         RETAILMSG(1, (TEXT("CMainWnd HandleVoice: Out of memory\n"));
```

```

2869c2884
<      pMainWnd->m_szAttachment[0] = '\0';
---
>      pMainWnd->m_fGotAttachment = FALSE;
2879a2895,2904
>      else if(pMainWnd->m_hWndVoice)
>      {
>          // We're in the middle of making a recording, so stop.
>          if(FAILED(hr = StopRecording(pMainWnd)))
>          {
>              RETAILMSG(1, (TEXT("CMainWnd HandleVoice: StopRecording failed %x\n"), hr))
>          ;
>
>              goto Error;
>          }
>      }
2882,2883c2907,2908
<          // We don't have an attachment so add one.
<          if(FAILED(hr = InsertAttachment(pMainWnd)))
---
>          // We don't have an attachment (and we're not in the middle of recording one),
so add one.
>          if(FAILED(hr = StartRecording(pMainWnd)))
2885c2910
<              RETAILMSG(1, (TEXT("CMainWnd CreateRichBody: InsertAttachment failed %x\n"),
, hr));
---
>              RETAILMSG(1, (TEXT("CMainWnd HandleVoice: StartRecording failed %x\n"), hr
));
2890,2891c2915
<          // Should have a name for the attachment by now.
<          ASSERT(pMainWnd->m_szAttachment[0] != '\0');
---
>          RETAILMSG(1, (TEXT("CMainWnd HandleVoice: Recording attachment now...\n")));
2968a2993,3064
> HRESULT CMainWnd::StartRecording(CMainWnd* pMainWnd)
> {
>     CM_VOICE_RECORDER vrec = {0};
>
>     HRESULT hr = E_FAIL;
>
>     lstrcpy(pMainWnd->m_szAttachment, TEXT("\\Windows\\Message.wav"));
>
>     vrec.cb          = sizeof(vrec);
>     vrec.dwStyle     = VRS_MODAL;
>     vrec.hwndParent  = pMainWnd->m_hWndNew;
>     vrec.id          = IDC_MSG_VOICE;
>
>     vrec.xPos = vrec.yPos = -1; // Center
>
>     vrec.lpszRecordFileName = pMainWnd->m_szAttachment;
>
>     if((pMainWnd->m_hWndVoice = VoiceRecorder_Create(&vrec)) == NULL)
>     {
>         RETAILMSG(1, (TEXT("CMainWnd StartRecording: VoiceRecorder_Create failed\n")));
>
>         goto Error;
>     }
>
>     // Kick of the recording!
>     ::SendMessage(pMainWnd->m_hWndVoice, VRM_RECORD, 0, 0);
>
>     // Everything worked ok.
>     hr = S_OK;
>
> Error:

```

```

>
>     return hr;
> }
>
> HRESULT CMainWnd::StopRecording(CMainWnd* pMainWnd)
> {
>     HRESULT hr = E_FAIL;
>
>     ASSERT(pMainWnd->m_hWndVoice != NULL);
>
>     // We don't have a good attachment yet.
>     pMainWnd->m_fGotAttachment = FALSE;
>
>     // From the spec, "Stop Record/Playback save recording data if any and close Voice
Recorder"
>
>     // Stop the recording!
>     ::SendMessage(pMainWnd->m_hWndVoice, VRM_OK, 0, 0);
>
>     // BUGBUG - How do we know if all that stuff worked?
>
>     // One would imagine that the voice control has destroyed the window.
>     pMainWnd->m_hWndVoice = NULL;
>
>     // Now add the attachment!
>     if(FAILED(hr = InsertAttachment(pMainWnd)))
>     {
>         RETAILMSG(1, (TEXT("CMainWnd StopRecording: InsertAttachment failed %x\n"), hr
));
>
>         goto Error;
>     }
>
>     pMainWnd->m_fGotAttachment = TRUE;
>
>     // Everything worked ok.
>     hr = S_OK;
>
> Error:
>
>     return hr;
> }
3018,3024d3113
<     if(FAILED(hr = GetAttachment(pMainWnd)))
<     {
<         RETAILMSG(1, (TEXT("CMainWnd InsertAttachment: GetAttachment failed %x\n"),hr))
;
<
<         goto Error;
<     }
<
3190,3205d3278
< HRESULT CMainWnd::GetAttachment(CMainWnd* pMainWnd)
< {
<     HRESULT hr = E_FAIL;
<
<     lstrcpy(pMainWnd->m_szAttachment, TEXT("\\Windows\\Message.wav"));
<
<     ASSERT(pMainWnd->m_szAttachment[0] != '\\0');
<
<     // Everything worked ok.
<     hr = S_OK;
<
< //Error:
<

```

```
<    return hr;  
< }  
<
```

```
#D    5487    384861
```

time	diff	comment	user	log.txt	op	path
Log for outlook\tinbox\app:						
██████████@14:49:16 (Wed)		addfile	GARRETT6			
██████████\STINGER\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	1				I1	
Application						
██████████@14:37:46 (Mon)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	2				I2	Use
custom draw listview						
██████████@13:03:07 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	3				I3	Bold
unread msgs						
██████████@13:28:29 (Thu)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	4				I4	Add
columns						
██████████@12:33:59 (Fri)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	5				I5	Add
syncing						
██████████@16:30:27 (Fri)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	6				I6	Add
temp width calculations						
██████████@09:03:24 (Mon)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	7				I7	
Improve temp scrollbar handling						
██████████@13:05:21 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	8				I8	Try
to send a msg						
██████████@12:50:30 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	9				I9	
Framework for SMS transport						
██████████@16:17:34 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	10				I10	
Temporarily regiser the sms transport						
██████████@18:04:00 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	11				I11	
Re-enable sip button						
██████████@16:16:38 (Fri)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	12				I12	Show
either read/unread icon						
██████████@11:47:13 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	13				I13	
Select first item						
██████████@12:39:29 (Wed)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	14				I14	Add
Rapier-style menu						
██████████@13:11:52 (Mon)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	15				I15	Add
recips to msg dlg						
██████████@16:25:40 (Mon)		in	GREGBURG12			
██████████\STINGER\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	16				I16	
Read Message Work						
██████████@14:07:34 (Tue)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	17				I17	
Change sms.dll name						
██████████@11:49:40 (Wed)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	18				I18	Add
startup msg						
██████████@16:09:42 (Thu)		in	GUYBARK4			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	19				I19	
Display Send window						
██████████@08:25:33 (Fri)		in	GUYBARK4			

log.txt

msg body	\\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 20	I20	Add
@13:03:52 (Fri) GUYBARK4 in	\\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 21	I21	Add
Contacts lookup	@12:13:28 (Sat) GUYBARK4 in	I22	Use
FileAs in New dlg	@16:33:56 (Sat) GUYBARK4 in	I23	RE,
FW and bodies	@18:20:02 (Sat) GUYBARK4 in	I24	
GuyBark: Add a simple Delete	@11:08:23 (Sun) GUYBARK4 in	I25	
GuyBark: Use RichEdit for body	@12:13:28 (Sun) GUYBARK4 in	I26	
GuyBark: Send dummy wav file	@13:02:58 (Sun) GUYBARK4 in	I27	
GuyBark: Size RichEdit wnd	@13:35:40 (Sun) GUYBARK4 in	I28	
GuyBark: Add GetAttachment	@18:08:59 (Sun) GUYBARK4 in	I29	
GuyBark: SMS Notify	@13:24:07 (Mon) GUYBARK4 in	I30	
GuyBark: Add body to notification	@14:19:39 (Tue) GUYBARK4 in	I31	Use
RichEdit	@11:40:40 (Wed) GUYBARK4 in	I32	First
pass at voice record	@16:17:26 (Wed) GUYBARK4 in	I33	Add
soft btn handling	@18:25:02 (Wed) GUYBARK4 in	I34	
Detect record failure	@08:43:05 (Thu) GUYBARK4 in	I35	
Change msg dlg look	@11:14:14 (Thu) GUYBARK4 in	I36	Tidy
SMS notification ui	@11:55:15 (Mon) GUYBARK4 in	I37	Tidy
up softkey use	@10:20:37 (Tue) UMESHMO3 in	I38	t9
support for send note	@13:50:17 (Tue) GUYBARK4 in	I39	Arrow
key changes focus	@11:58:57 (Wed) GUYBARK4 in	I40	Move
caret in edit ctrls	@11:17:07 (Thu) GUYBARK4 in		

log.txt

██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	41	I41	
Explicit windows path for dll			
██████████@13:12:42 (Fri) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	42	I42	Make
msg wnd modal			
██████████@12:20:32 (Sat) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	43	I43	Mody
demo ui tidy up			
██████████@12:56:40 (Sat) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	44	I44	Find
phone# thru e-mail address			
██████████@11:13:26 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	45	I45	Don't
drop the dlg			
██████████@12:53:40 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	46	I46	
Reenable owner draw list			
██████████@13:21:14 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	47	I47	
Delete existing wav file if necessary			
██████████@14:00:46 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	48	I48	Don't
invoke menu if menu already visible			
██████████@14:17:31 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	49	I49	More
menu handling			
██████████@14:43:47 (Sun) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	50	I50	Show
hourglass during sync/send			
██████████@11:33:23 (Wed) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	51	I51	Add
DEMO cmd line arg			
██████████@12:20:20 (Wed) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	52	I52	Set
focus when become foreground wnd			
██████████@09:15:50 (Thu) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	53	I53	Stop
flash during T9 entry			
██████████@09:56:46 (Thu) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	54	I54	Check
for existing notification wnd			
██████████@12:49:10 (Thu) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	55	I55	Set
focus when dlg dropped			
██████████@09:31:50 (Fri) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	56	I56	Don't
cancel the recording			
██████████@16:31:10 (Fri) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	57	I57	Check
reg content for SMS class			
██████████@10:32:45 (Mon) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	58	I58	
Remove not-required reg of SMS			
██████████@14:36:45 (Mon) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	59	I59	Allow
TALK from SMS notification			
██████████@17:48:15 (Mon) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	60	I60	Allow
smsmsg file number			
██████████@16:52:24 (Tue) GUYBARK4 in			
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp	61	I61	More
demo related changes			
██████████@11:23:52 (Wed) GUYBARK4 in			

log.txt

██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 62	I62	Show
T9 text between btns		
██████████@13:25:32 (Wed) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 63	I63	Make
text big and bold		
██████████@11:34:22 (Thu) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 64	I64	
change dlg appearance		
██████████@08:56:18 (Fri) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 65	I65	Close
menu before notification		
██████████@10:19:24 (Fri) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 66	I66	270:
Don't add Reply body		
██████████@11:04:35 (Fri) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 67	I67	
Disable Options and Send for demo		
██████████@16:48:57 (Sat) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 68	I68	More
T99 demo fixes		
██████████@11:30:34 (Tue) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 69	I69	Don't
setfocus to msg list if notif around		
██████████@16:29:12 (Wed) GUYBARK4 in		
██████████\TITANIUM\PRIVATE\APPS\OUTLOOK\TINBOX\APP\main.cpp 70	I70	
Always allow TMail to be brought forward		



[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

Remarks by Bill Gates Microsoft Corporation

Redmond, WA

Bill Gates' keynote is available for
viewing via Windows Media:

- 28.8K
- 100K

MR. GATES: Welcome [REDACTED]. There's more going on with communications companies today than with any other kind of partner. And the kind of empowerment we'll be able to provide to workers around the world by bringing software and great communications together is very, very exciting.

Now, the vision of Microsoft originally talked about, a PC in every home and on every desk, that vision has made incredible progress. But in some ways, even that breathtaking vision doesn't encompass what's going on today. So, for the first time in our 25-year history, we actually changed our vision statement to talk about empowering people in an even broader fashion: any time, any place, on any device. Making it so that they don't have to think about moving their information around. Any files or favorites or messages that they're interested in should just immediately show up wherever they are, whether it's the TV that will be connected to the Internet, their mobile phone, their computer in their car, or their PC in all its various forms.

Now, in order to make this happen, we are completely dependent on forming strong partnerships, partnerships with telecommunications companies. And that's why, in the last year, not a week has gone by without some new announcement of how we're working with a great telecommunications partner or bringing software to provide new services for end users.

This combination, I'd say, is very similar to the collaboration that existed between the hardware industry and Microsoft in creating the PC business. Companies like Intel and Compaq were the key partners in those years. Today I would say that our work with telecommunications companies is, if anything, equally important to those kinds of partnerships.

So what are we trying to do for people? Well, we're trying to make it so they have new forms of communications. They can stay in touch with people, know the status of people, meet with people in far more efficient ways than ever before.

Our role as a software provider is to take this idea of the PC, where people are building digital documents, and extend that, whether it's digital photography, digital music, video conferencing, stored digital meetings. We'll take all of those things and extend them out to all the ways that people access communications. And so that's why you see partnerships for broadband communications.

Getting the bandwidth up is very, very important. And certainly the business-to-business connectivity is moving at an incredible pace. The one place that is still a challenge is getting high-speed connections into every home. I believe that DSL will explode in the years to come, even though the volumes today are quite modest, and we'll continue to see phenomenal growth in cable modems. Those will be the primary ways that we reach out to homes as well as businesses and give people access to all this information.

We have a number of partnerships aimed at taking our software and applying it to that broadband environment. Some of the applications, I think, aren't anticipated. For example, we're able to monitor someone's PC and support them in a whole new way. If they need advice, if they want help on anything, by using that high-speed network we can immediately look at their screen and guide them to the option that they need. And so the whole notion of support and reliability

are raised to a new level with that broadband connection.

Other partnerships have to do with mobile data services. It's an exploding area, but it's going beyond just voice to having access to the Internet itself through the mobile phone. And bringing together these standards, the Internet, XML standards, the wireless standards, that's one of the roles we have, working through the standards groups.

Finally, we have the idea of hosting; communications providers providing far more than just the bandwidth, actually bringing the servers -- whether it's mail servers, file servers or application servers -- to their customers, and so being on a high level on the value chain. We see a big opportunity working with communications companies to make that a reality.

Knowledge workers are working in a different way today than in the past. The idea of electronic mail is really taken as a starting point, and the kind of advanced collaboration or learning that we'll have on top of that is already being shown off in pioneering companies. So the portable PC has come down in price. The volumes have exploded, along with a wide range of software that runs on a PC that's sort of a standard tool for knowledge workers.

Now, we can take all of that and extend it across all these different devices. And thinking through the scenarios for the knowledge worker is one of the things we do best; for example, making it so that all your different messages show up in one place -- voice mail, fax -- so that if you define groups of people, your buddy list or your address groups, that those are available automatically on all these devices. Those are the things that we think knowledge workers are going to be very excited about and far more productive as they become available.

The PC continues to surprise the analysts. Each year for the last four or five years, people have had very conservative forecasts for the PC, saying that maybe this was the year that PC sales would actually be down. In fact, the installed base of PCs has grown very, very rapidly. Not only has the price gone down and the volume gone up; the actual power of the device continues to increase; things like the expandability through the universal serial bus, more and more use of flat screens that will have the resolution to allow us to move to a paperless environment, more and more wireless connections so you'll be able to take that PC and carry it around your business, or even out into the world at large, and still have access to new messages or data changes that you're interested in.

So the PC is not to be underestimated. The volume today is well over 100 million units a year, and that's driven the software industry to create some very, very exciting things.

However, when you think about your PC today, you still have too many things that you have to move around yourself. If you have two PCs and are moving the files back and forth, it's a lot of work. Making sure that when you go home, you have access to the information you had at work; that requires special steps.

What we have to do here is work with telecommunications providers so all that information -- business databases, files, contacts -- all of it shows up automatically as soon as you authenticate who you are, then your personalized desktop, that shows you the part of the Web or your mailbox that you care about, that immediately comes up on the screen, including adapting to the screen size of the device that you're looking at.

So we have quite a range of devices that are part of this information any time, anywhere. We've got the full-blown PCs that have the screen size that lets you create and annotate documents. This idea of annotating and collaborating is also exploding, whether it's voice notes or handwritten annotations. The ability to share those across the Internet with all your co-workers is an important productivity improvement.

Now, these intelligent appliances, whether it be the TV sets or digital cameras or high-end copiers, there's embedded software that needs to connect up, use the same standards and understand what the formats look like for all of the different information. And so by taking rich software technology, those devices become far

more capable.

At the low end, we have very simple devices like a light switch or a pager or a simple phone with just a microbrowser, but we need to do the very simple layer of software, relying on the servers to do the actual hard work. And so this is a new area for us, including the smart card capabilities that are being offered now for the GSM phones or for PC authentication.

Now, each of these devices will connect up to the Internet and get rich services. Authentication is the first one that's necessary, but beyond that you need directory capabilities, unified messaging capabilities, all of which run on the Internet and the user doesn't have to think about where those things actually are.

Now, we believe these scenarios will excite users, and so that in the same way the PC took off by word of mouth, the demand for these things will really explode without needing a lot of marketing. It requires end-to-end solutions. It requires integrated solutions. Some communication centers today, if you work with them on the mobile side, you get one mailbox. If you work with them on their broadband capability, you get another mailbox. And that's a different mailbox than the one you may have at home and different than the one you have at work. And so you're forced to be the coordinator of all those pieces.

If we can get rid of that and make it seamless to see the information on the mobile phone or the new devices that'll have even larger screens, that's where we'll really see the demand take off and see the full potential of information any time, anywhere.

We're very pleased to see the evolution taking place of the wireless standards and the Internet standards. Bringing those together has been a mammoth task over the last few years, and we're getting really excellent standards for the wireless connections; 802.11 and so-called "Bluetooth" will be built into many of these devices, and so they'll automatically discover what services or other devices are around. If you take your phone into your home environment, you'll even have advertising on the wireless network that will say there's a TV there you could control or you could select music and direct that to any of the speakers in your house. And so having even the phone itself be a control device in these wireless networks is something that the higher-level protocols will guarantee.

Well, I thought it'd be fun to show you an example of how this can come together, and so I'd like to ask Ian Ferrell, our program manager for our smart phone effort, to come up. And what we're going to show you here is how we've taken Microsoft Outlook, which is part of our Office suite, and the Internet Explorer browser and actually built it into a new mobile phone.

MR. FERRELL: Hi, Bill. This is a prototype GSM smart phone that we built back at the Microsoft lab. It is a Windows CE-based device that has an integrated GSM radio module. And if you like, I can show you how it works.

MR. GATES: Yeah, let's see if it works. (Laughter.) That's a beautiful screen.

MR. FERRELL: Yes, this is a color screen. People definitely like the high fidelity on these smaller devices. What we have here is kind what you see with a traditional cell phone, the carrier information with the battery and signal strength, but there's also additional information that's on this device, such as mail messages that have come down while I've been standing back there, and also an upcoming appointment that I need to be aware of.

We also have a start menu here with these smart buttons. The start menu will allow me to access my applications. And then, over here on the options button, I can bring up the phone option.

So actually right here what happens is we just got a message alert. What we've done is with the tie-in with exchange, E-mail is sent directly to the device. I don't have to worry about having multiple E-mail accounts at this point. The E-mail is broadcast directly to the phone. And with extensions that are available in the Exchange Server coming up, the E-mail is sent down automatically.

MR. GATES: So that person didn't have to think about sending to your mobile phone. They just sent to your normal address, and you set up rules that controlled which messages would be passed to the phone.

MR. FERRELL: Correct. When you send a piece of E-mail to somebody, you don't have to think, "Now, am I sending it to their work? Am I sending it to their phone?" You just send it to them. This device becomes kind of just another Outlook client, another window onto your Exchange information.

MR. GATES: Now, let's say I want to respond to this. What kind of options do I have?

MR. FERRELL: You have a lot, actually. Let's open up the mail message here. Some of the quick options that I have is I could certainly reply back and send a text message if I wanted to. Another quick way, since this is a phone, I could just press the talk button and call the person directly.

MR. GATES: Instant messaging.

MR. FERRELL: Instant messaging. There are some other options as well. I could just reply back with a voice message if I maybe didn't want to disturb them, or I could send them a fax message.

MR. GATES: Okay. So if I send a voice message, if they're running on their PC, how will that show up? Will they be able to play back the voice?

MR. FERRELL: Yes. It'll show up just as a .WAV attachment that they can double-click on, and the information will be played over their speakers. We could try it. Would you like to see?

MR. GATES: Okay, let's give it a shot.

MR. FERRELL: I'm just going down here and selecting "Reply by voice." The mail message was asking me where we're going to have food for our meeting after this. I'm going to look for some food and I will let you know the location as soon as possible. So here we've put a .WAV attachment onto the mail message. A lot of times when you're with a phone, you don't want to spend a lot of time entering text. It's just much quicker to make a quick voice recording and send that off to the user. So you can just press the send button and the message goes off. This is actually being synchronized in the background back to my Exchange server, and then it goes back out over the network.

MR. GATES: Well, and do we have access to the Internet here? What kind of information services can we get to?

MR. FERRELL: We can definitely get to the Internet. In fact, so that question wanted to know where we can get some food. So let's see if we've got a Web site we can go to. Let's go to a Web site that our MSN mobile team has set up. This is some city information on Geneva. So what we're doing right now is creating a dial-up connection to the Internet and downloading a Web page with some Geneva-specific information onto the phone.

MR. GATES: Now, this Web page can use just standard HTML. It doesn't have to be a different offering environment. It's the same as the normal Internet.

MR. FERRELL: Correct. This is a traditional Internet Explorer. It's got all the full Internet standard support like HTML and scripting. And this is a plain-Jane HTML page.

MR. GATES: I see you've got accommodations there. I don't think we need to look there. We know there's nothing available within about 50 miles. (Laughter.) So let's try restaurants.

MR. FERRELL: Okay. So what we'll do is we'll go to a restaurant page that will allow us to search for a restaurant of our choice. Let's actually get some Italian food. And would you be willing to pick up the check?

MR. GATES: Sure.

MR. FERRELL: Okay, we'll go four-star then. (Laughter.) What this is doing is it's doing an over-the-Internet connection back to a SQL database. And based on our criteria, they're returning a list of actual restaurants that match that. The little icons off here to the side are giving me some additional information on these restaurants, such as whether I can dial them directly from the phone, whether there's a map available. This first entry here, the Trattoria, has an icon that means that they support E-commerce.

MR. GATES: Fantastic. Let's give it a shot.

MR. FERRELL: So what we'll do is we'll connect to the Trattoria Web site. Again, the restaurant, it could have a Web site available for desktop users and the code would be fundamentally the same, pure HTML.

MR. GATES: So you even have, for example, Jscript in that page, all the same kind of richness that you're used to on the PC.

MR. FERRELL: Absolutely. And rich graphics, as well with this nice color map. So if I was going to walk there or my cab driver thinks he needs to go through Milan to find this restaurant, I can find out, using the map. And then down here we have a couple of links. The first one I could actually dial them up and call. But that's a little low-tech, so let's try buying something actually over the Internet. So this will actually be entering the E-commerce portion of the site.

This page here actually uses some client-side scripting to update my order and to put together the order. What it is doing is it's using the actual desktop scripting that you would use for an ordinary E-commerce site. So if you do have an E-commerce site set up, you don't have to go back and learn new tools and new traditions and ways of doing things, you can just use your existing technologies.

So let's place that order. Again, we're going back over the Internet and booking this. So we've got an order confirmation back, and the Microsoft wallet on the phone has thoughtfully put in my credit card information. And because I've been here before, we're using some of the auto-fill to fill in things like the delivery address. So I have to input the minimal amount of data that I need.

MR. GATES: Now, that credit card. What kind of security problem would you have if you send that across the wireless network?

MR. FERRELL: You would definitely want to be concerned. But in this case, the Web browser supports SSL. So we're in a fully secure end-to-end Web transaction at this point. The same thing goes for corporate data, if you're accessing the corporate or website over the Internet here. Because the browser supports SSL, you're well protected.

So let's actually submit this order and get our pizza on the way. And so here we've gotten back a confirmation. They've gotten the order and they'll be delivering the pizza, and not a moment too soon, too, because I missed breakfast today.

Another thing about this is that it is a personal device that you'll be carrying around with you, so you'll want to carry some subset of the data that you normally have in Exchange today. We do have support for things like your contact management. I have a couple of thousand contacts, all the business cards, the people that I've met over the years. I can take their phone numbers, their mailing address and their E-mail address and other information along with me. And if I make any changes on the device, much as the E-mail is automatically transferred in the background and I don't need to press any buttons, these changes are automatically updated as well.

The same thing goes with the calendar. I can take calendar appointments with me. It will notify me when those appointments come due. And I can either be buzzed or the little vibrator can run for me. And any changes I make there or any requests that come in, I can book automatically.

MR. GATES: That's fantastic. It's great to see it all running on a mobile phone.

MR. FERRELL: Absolutely.

MR. GATES: Thanks, Ian.

MR. FERRELL: Thank you. (Applause.)

MR. GATES: Well, that's a level of power that even the PC itself didn't have a few years ago. And yet by working with these telecommunications companies, that's something that I think millions of people will be taking advantage of in the next year.

Well, one of the advances we've been working hard on is the Windows platform itself. We use the same technology, Windows 2000, for both server and client. And this Windows 2000 release, which is very, very soon, is something we're really pleased to see come together, because it has all the work, all the feedback we've gotten in the last few years, making it very manageable, making it a great hosting server for even the most demanding applications. We even have people that are going to be hosting thousands of Web sites on a single machine.

Now, people are very interested in what kind of scale capability this can provide. And people have no doubt that the PC server is the place where you get the widest breadth of applications, you get the best price performance, you get the best integration with the PC and other knowledge worker devices. But the one question they're always left with is what about scale? After all, on the Internet it's hard to predict the demand you're going to have. And having your Web site stay up 24 hours a day and be totally reliable and have high responsiveness is very important.

Well, there's two different strategies for attacking the scale challenge. One is to build bigger boxes, and that's called "scale up." In fact, the progress there is pretty fantastic. You know, we've gone from 2,000 transactions a minute to over 40,000 transactions a minute. In fact, that can handle the enterprise resource planning software for even the biggest businesses.

But with the Web we're going to have to go even further. In fact, this scale up approach with faster chips and faster buses over the next 18 months will more than triple in speed. So we'll go up over 100,000 transactions a minute.

So what that means is that the PC environment will match the performance of even the most expensive mainframe or UNIX environments.

But as great as that scale up approach is, using multiple processors, going up to 32 or 64 processors, it does have a drawback. And that is that it doesn't let you buy exactly the amount of computing you want, and, most importantly, it becomes the single point of failure.

And so while matching there is definitely part of our strategy, the thing we'll provide that's very, very unique is this ability to scale out; that is, to add individual servers exactly as you need them. And what happens is that the load against your Web site is balanced across all those servers.

Now, in order to make this work you have to transact all the activity so that if a machine ever breaks down, loses power, that that work can be transferred without any difficulty to the other machines. And so that's why we've built the transaction capability into the heart of the operating system.

So this scale out approach has some incredible advantages: modularity, low price starting point, lets you go up to arbitrary levels. You know, you're always going to have limitations to scale up, because the bus speeds are going to be the limiting factor. But with scale out, where you balance the load across hundreds or even thousands of these machines, the sky is the limit.

The one challenge here is making it easy to manage the different systems, making it so you can update the software and check what's going on all in a very easy way. And that challenge has to be met by having more advanced software. And, in fact, with Windows 2000 our apps center server is aimed at doing that.

Well, I mentioned earlier that Telecom 99 is very well-timed, because it lets us herald the importance of all these different partnerships we're putting together with communications companies. I've shown some of the logos. I've shown some

of the headlines of what's going on here. What this represents is a belief by Microsoft that the demand for advanced communications has been greatly underestimated, whether it's broadband out of the home, whether it's getting data onto your mobile phone. We've made a bet that these things will really explode. And we've taken our software R&D budget and really aimed at these integration scenarios, integrating the TV, the mobile phone and the PC together, and letting people get at information in a way that they don't even have to think about where that's located.

In each of these partnerships we're taking our technology and the network and services that the partner once provided and coming up with new offerings. And you see this is on a very global basis. You know, ever continent we have different things that we're doing, but it all comes together to really drive things for that knowledge worker.

One of the things we've also been able to do is help accelerate some of the deployment here by putting in our R&D dollars, by doing special software design, and by providing in some cases investment money, we help accelerate the vision that we have here.

Now, it's important to note that we stay focused on what we're good at, which is building software. Any of the investments we do are minority investments. We don't seek any kind of controlling position, because we want to get up every day and think about software, think about the horizons out there; for example, the voice recognition that will be part of those servers and make that mobile phone scenario even more exciting. For example, the tablet PC, a PC without a keyboard that you can take into a meeting and with the wireless connection, any of the notes, the handwriting that you do on that tablet, that the software will recognize, can be immediately shared throughout your enterprise.

And so tackling those tough software problems is the thing that has caused us to continually increase our R&D budget to now over \$3 billion a year.

But even doing great software doesn't make a full solution unless we focus on these partnerships. And that's a big reason why we're here at Telecom 99, highlighting that progress, highlighting our commitment to work with these telecommunications partners.

It's really very exciting what can be pulled together here. The way to think about this is to think about yourself, and today how you deal with information. You know, with business data, how do you track the accounts that need more attention? How do you track project status? Could you easily do a videoconference with people where you pull together and look at Windows applications and talk about what's going on, where you're all editing those things together?

Well, in fact, most of those scenarios still are not achievable. And so there's a whole new way of working here that can be brought about. That's the demand that justifies the very high levels of investment we see today; investments in communications equipment and infrastructure, investments in start-up companies, and so it's a really amazing time to be part of these activities. And we look forward to seeing these solutions rolling out as we work with our partners.

Thank you.

(Applause.)

Bill Gates' keynote is available for viewing via Windows Media:

- 28.8K
- 100K